Model Based Off-Line Method for Velocity Trajectory Compensation

Shital S. Chiddarwar
Department of Mechanical Engineering
Visvesvaraya National Institute of Technology Nagpur
Nagpur, India
s.chiddarwar@gmail.com

***Abstract*— The velocity compensation method proposed in this paper computes the value of velocity compensation in the off-line mode to compensate the effects of non-linear dynamics of the robot. This paper explains a methodology that utilizes a priori knowledge of the robot dynamics to improve the path tracking accuracy by generating an additional velocity compensation based on robot dynamics. The strategy for off-line velocity compensation entails the determination of inverse dynamic model of the robot system. This model consists of inverse dynamic model of robot and the inverse model of the controller. The inverse dynamic model of robot is highly nonlinear where as controller model is linear. Both these models need to work in closed loop that demands for linear presentation of the robot inverse dynamics. In this work, a body oriented method for linear form of dynamics equation and PI controller model will be used. The proposed velocity compensation algorithm will be demonstrated for a spray painting robot using SIMULINK environment of the MATLAB. The results will be evaluated at joint as well as Cartesian level. Finally, the path tracking error with and with considering compensation will be determined for the case problems.**

Key words: path tracking; spray painting robot; inverse dynamics; trajectory compensation

I. INTRODUCTION

Industrial robots have become an indispensable means of automation to increase productivity and flexibility of production systems. The ever-increasing quality standards and international competition imposes higher requirements on reliability, positioning accuracy and velocity of industrial robot. Moreover, modern applications like laser cutting, welding and water jet cutting require an increasing path tracking accuracy. For applications like spray painting, it is necessary to move the end-effectors of manipulator along some desired paths with prescribed speed [1]. Path tracking errors mainly originate from kinematic errors, controller performance limitations and joint flexibility [2]. The kinematic errors can easily be compensated in the path planning and flexibility is mostly negligible because commercial robots have high transmission stiffness [3]. Typical industrial controllers are capable enough to deal with simple pick and place sort of application. However, the controller required for high-speed path tracking should take into account nonlinearities like centrifugal, gravitation, coriolis forces, friction, actuator dynamics and dynamic coupling between axes, resulting in deviations from the desired motion [4]. Since there is a rather slow evolution to change the standard controller for industrial robot, a compensation of the non-linear dynamics can be realized by adding compensation to the desired trajectory [5]. The soft computing approach i.e. application of neural network controller for industrial SCARA robot was attempted. This attempt was fusion of two controllers, one PD controller for velocity compensation and another NN based for computation of torque/forces [11]. Another related work is by Chin Su Kim and Kang Woong Lee [12]. They have developed a new robust feedback controller for trajectory control of n-link robot manipulators with parametric uncertainties. Their proposed controller using dynamic compensation scheme leads to improvement in transient response while achieving asymptotic regulation in the presence of parametric uncertainties. The experiments were performed on a 2 dof robot. On the similar track, the problem of output feedback tracking control of robot arm modeled as an Euler-Bernoulli beam was addressed [13].

This paper explains a methodology that utilizes a priori knowledge of the robot dynamics to improve the path tracking accuracy by generating an additional velocity compensation based on robot dynamics. This can be seen as off-line compensation of the velocity trajectory in such a way that after execution the end-effector follows desired position trajectory more accurately. The strategy for off-line velocity compensation entails the determination of inverse dynamic model of the robot system. This model consists of inverse dynamic model of robot and the inverse model of the controller. The inverse dynamic model of robot is highly nonlinear where as controller model is linear. Both these models need to work in closed loop which demands for linear presentation of the robot inverse dynamics. In this work, a body oriented method for linear form of dynamics equation and PID controller model will be used. The proposed velocity compensation algorithm will be demonstrated for a spray painting robot using SIMULINK environment of the MATLAB. The results will be evaluated at joint as well as Cartesian level. Finally, the path tracking error with and with considering compensation will be determined for the case problems.

II. PROPOSED APPROACH

The perfect path tracking for an industrial robot that is a complex dynamic system is not easy to realize. The classical proportional controller with feedback always lags behind in executing the desired trajectory [6]. This classical controller utilizes position and velocity tracking error to build up the required torque. As a result, a large tracking error is necessary to generate high actuating torques. The proper remedy to this would be an addition of a torque feed forward. This will be responsible for determination of required torque and hence the proper torque generation. However, almost no industrial controllers will allow intervention up to this level of control architecture. Hence, it is necessary to make a feedback controller, which will generate desired torque. In other words, the position command responsible for generation of desired actuator torque should be estimated. The proposed approach for velocity trajectory compensation is as presented in Figure 1. In order to implement the proposed methodology, a model of the inverse robot system has to be identified. The inverse model of robot system consists of two vital components: an inverse model of dynamics of robot and an inverse model of the controller. For this work, Kuka Kr-15 robot is considered. Its dynamic parameters and inverse dynamic equations are obtained from the doctoral thesis of the author [6].

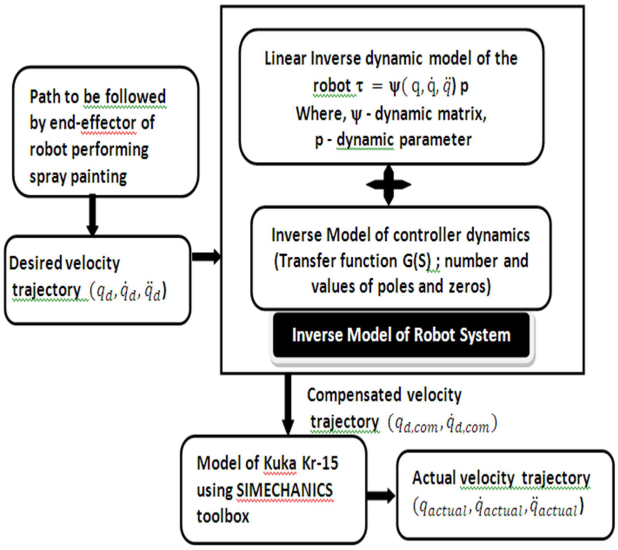

Figure 1 Schematic of proposed approach for velocity trajectory compensation and tracking

a) Inverse dynamic equation for Kuka Kr-15 in linear form :

The dynamic model of robot describes relation between the robot motion and the required actuator torques. In includes non-linearties like friction, centrifugal, gravitation and Coriolis forces. The inverse equation of robot motion can be written in a linear form (equation 1) which forms the basis for an accurate interfacing with linear controller dynamics [7].

$$\tau = J_\tau^{-1}[J_1^{-T}\,\Omega_1 \quad J_2^{-T}\Omega_2 \;\ldots\ldots\ldots\; J_n^{-T}\Omega_n]\begin{bmatrix} p1 \\ p1 \\ . \\ . \\ pn \end{bmatrix} \ldots (1)$$

$$where, \Omega i = \begin{bmatrix} {}^{O}_{Ai}R & 0 \\ 0 & {}^{O}_{Ai}R \end{bmatrix}\begin{bmatrix} {}^{Ai}a_i & \Lambda_i & 0 \\ 0 & -{}^{Ai}a_i & \Psi_{wi} \end{bmatrix}$$

$$with\ {}^{Ai}a_i \equiv {}^{Ai}\ddot{x}_i - G; \quad pi = [mi, mir_i^T \; I_{pi}^T]^T$$

$$J_\tau \equiv [\, J_1^T\gamma_1 \, J_2^T\gamma_2 \ldots\ldots\ldots . J_n^T\gamma_n \,]$$

Where n = number of joint actuators

$$\gamma_i = \begin{bmatrix} 0 \\ 0 \\ \epsilon i \\ 0 \\ 0 \\ \bar{\epsilon_i} \end{bmatrix} \text{with } i \equiv \begin{cases} 0\ for\ revolute\ joint \\ 1\ for\ prismatic\ joint \end{cases};$$

$$\bar{\epsilon_i} = 1 - \epsilon i$$

Where, J_i is the Jacobean matrix for link i ; ${}^{O}_{Ai}R \rightarrow$ rotation matrix of link i w.r.t. base A; ${}^{Ai}a_i \rightarrow$acceleration of link i w.r.t. base A; $\Lambda_i, \Psi_{wi} \rightarrow$matrices containing angular velocity and angular acceleration of link i w.r.t. base ; m→mass of the link; mr→first order moment vector; Ip→ second order moment vector (moment of inertia). After substituting the numerical values of various parameters[5][6], the linear form of dynamic equation gave output in the form of desired torque for desired joint angle trajectory.

b) Inverse model of controller dynamics

The converse model of controller dynamics is obtained by considering one PID controller dedicated to one joint of Kuka Kr-15 robot. Since the robot has six joints, total six PID controllers are considered. Separate controller is necessary because the robot can move all six joints simultaneously. These six PID controllers are arranged in parallel to form the controller architecture for entire Kuka-Kr15 robot. The transfer function for PID controller can be written as

$$G_{PID}(s) = \frac{M^2(K_D s^2 + K_p s + K_I)}{Js^3 + (M^2K_D + b)s^2 + M^2K_{P}s + M^2K_I} \quad ;$$

Where, J → effective moment of inertia;
b→coefficient of viscous damping ;
M→gear reduction ratio from motor shaft to load shaft [10].

In order to find the gain values of the PID controller, each link of the robot is considered as a spring mass damper system. The transfer function for such system requires information regarding mass of the link, damping coefficient and stiffness. The numerical values of these parameters are given in Table 1.

Table 1: Parameters required for transfer function [5]

Link	1	2	3	4	5	6
Mass (kg)	0	33	32	19	9	15
Damping coefficient (Nm/s)	99.1	121.8	58.8	32.5	10.4	3.6
Stiffness (N/m)	0	1190	549	2303	357	357

Using the numerical value of these parameters, optimal PID gains are obtained using control toolbox of MATLAB. Automatically choosing optimal PID gain is possible using a tuning algorithm i.e. either 'pidtune' function or a nice graphical interface 'pidtool'. In this work 'pidtool' is used which chooses PID gains to balance performance (response time, bandwidth) and robustness (stability margins). The phase margin is considered as 60 degree, response time of 2.5 seconds and Bandwidth of 32 rad/s. The PID gains obtained using automatic PID tuner 'pidtool' and these parameters are given in the Table 2.

Table 2 : PID gains for six PID controllers

Link	1	2	3	4	5	6
Kp	55	125	65	52	23	7
K_D	25	63	38	28	11	4
K_I	45	78	25	49	13	2

These parameters are used to obtain the transfer function for each PID controller. All six were arranged in parallel to get complete controller and further its inverse was taken. This model was developed using simulink toolbox of MATLAB. The desired torque coming as an output of the inverse dynamics model of the robot is given as an input to this model and output is the compensated velocity profile.

c) SimMechanics model of Kuka Kr-15

The mechanical model of Kuka Kr15 was generated using SimMechanics toolbox. The mechanical model for first two joints of the robot is given in the Figure 2. On the same lines, mechanical model remaining four joints is also made. Due to space constraint, the complete model for the robot cannot be accommodated here (6). The input to the model is the torque required and output is the actual velocity of the joint.

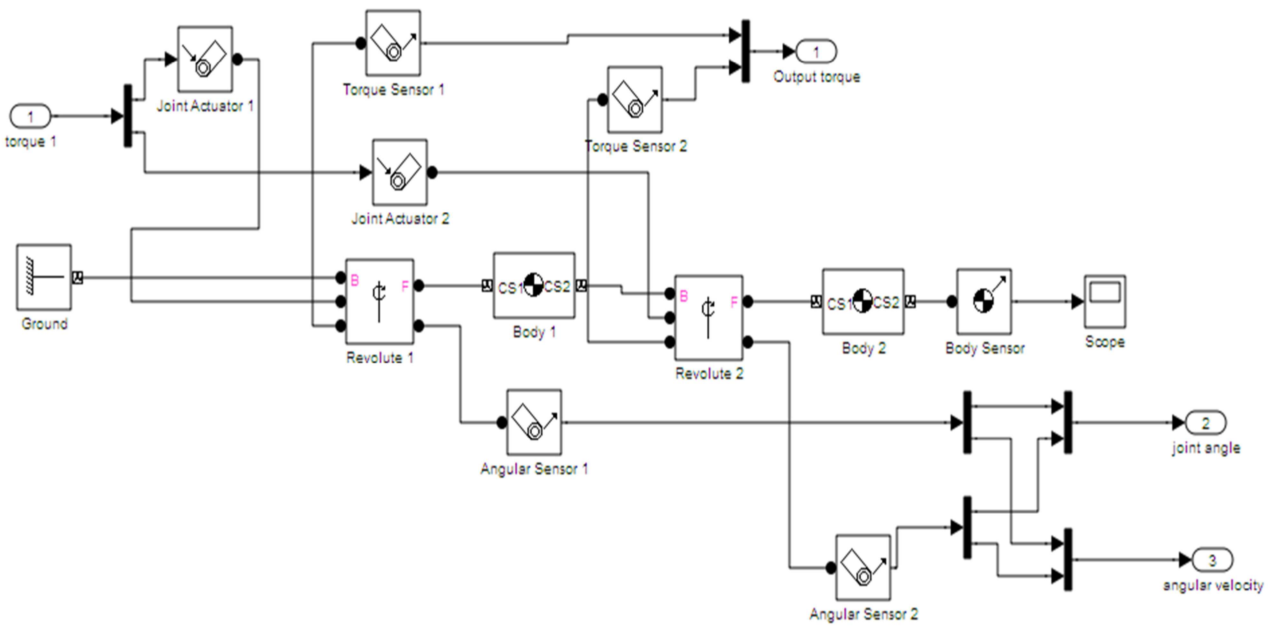

Figure 2 Mechanical Model of two joints of robot using SimMechanics toolbox

III. Working Principle Of Proposed Approach

The input to the complete system is the desired joint angle trajectory. This trajectory is given as an input to the inverse dynamic model and a derivative block. The inverse dynamic model explained in the section 2.1 obtains the torque required for motion of each joint of the robot along the desired joint angle trajectory. The derivative block gives desired velocity of the joint. The torque this obtained is given as an input to the inverse model of controller dynamics that carries out compensation and gives compensated velocity as an output. Since standard industrial controller cannot be changed to provide a computed torque feedforward, the trajectory precompensation is necessary to be added using the available velocity feedforward controller input. This compensated velocity is given to the velocity controller which is nothing but the proper model of the controller dynamics. This in turn gives an output as a torque. This torque is further given as an input to the robot block that is a SimMechanics model of the Kuka Kr 15 robot. The output of this robot block is the actual velocity of the robot joints. An integral block obtains the actual joint angle position for the joints of the robot. The actual joint velocity and joint position are fed back to the velocity controller and position controller. This is necessary to fine-tune the values of angular velocity of joint and the position of the joint. Finally, the numerical values of desired and actual joint angle positions are compared to find out the tracking error.

IV. Sample Example

The proposed strategy is implemented to a spray painting operation. The generation of accurate tool / spray gun trajectory and its execution is most important since uniform coating of paint and optimal paint utilization are dependent on them. In spray painting operation, trajectory to be followed by spray gun is decided by the model of the spray gun and the topography of the surface to be painted. The trajectory obtained from this is in the Cartesian form and must be converted into joint angle trajectory using inverse kinematics relations of Kuka-Kr15 robot [9]. In this case study, it is assumed that the joint angle trajectory required to generate a 'S' shape trajectory on the horizontal plane is available. This shape is generated using velocity of 600 mm/s. the study of only first three joints is involved in this case because for spray painting, the orientation of the spray gun is kept so that it remains perpendicular to the horizontal surface. The desired joint angle trajectory 'q_d' is given as an input to the proposed system to get the 'q_{actual}' is obtained. The tracking error q_{d-} q_{actual} with ans without compensation of the non-linear dynamics executed for the S-shaped trajectory in a horizontal plane. The corresponding maximum absolute values of tracking error are given in Table 3.

Table 3 Absolute values of tracking error

Joint	Without compensation	With compensation
Joint 1	3.461313 x10^{-3} rad	0.591258 x10^{-3} rad
Joint 2	1.79 x10^{-3} rad	0.338911 x10^{-3} rad
Joint 3	5.381382 x10^{-3} rad	0.772291 x10^{-3} rad

To have measure of the improvement of the absolute path tracking accuracy, Cartesian positions corresponding to 'qd' and 'qactual' were computed using forward kinematics of the Kuka-Kr15 robot [8]. The distance between the point (x_{actual}, y_{actual}, z_{actual}) actual trajectory and the point (x_d,y_d,z_d) desired trajectory in Cartesian coordinates is given by

Euclidean distance di=

$$\sqrt{(\mathrm{xd(i)} - \mathrm{xactual(i)})^2 + (\mathrm{yd(i)} - \mathrm{yactual(i)})^2 + (\mathrm{zd(i)} - \mathrm{zactual(i)})\hat{}2}$$

To evaluate the performance of the proposed approach, following criteria have been considered

1. Mean deviation from desired trajectory $dmean = \frac{1}{N}\sum_{i=1}^{N} di$; N→number of measured points
2. Maximum deviation from the desired trajectory $dmax = \max di$

Table 4 shows the values for these performance criteria, and confirm the improved path tracking accuracy by using trajectory precompensation.

Table 4 Performance criteria for 'S' shape with 600 mm/sec velocity

	dmean(mm)	dmax(mm)
Without compensation	1.19	2.505
With compensation	0.29	0.7431

V. Results And Discussion

The proposed approach of trajectory precompensation was implemented to a case problem and results thus obtained are encouraging. The precompensation technique helps to reduce the tracking error which is useful to move spray gun along desired path with desired velocity. The performance of the proposed approach was tested on joint angle as well as Cartesian trajectory. From Table 3 and Figure 3, it can be seen that with compensation there is a significant improvement in the performance of the controller. The values of tracking errors by considering velocity compensation are between 0.33 x10^{-3} rad and 0.591258 x10^{-3} rad for joints under consideration whereas these values are more if compensation is not considered [1.79 x10^{-3} rad - 5.381382 x10^{-3} rad]. When compensation is considered there is considerable decrease in the error. This decrease in error improves the accuracy of path tracking for the robot. The reduction in error is around 50-70 % as compared to tracking error without considering compensation. However, it is necessary to have all dynamic parameters of the robot to accurately model it using SimMechanics and SIMULINK.

The results presented in Table 4 indicated the mean and maximum deviation of the obtained trajectory from the desired trajectory. These results were obtained for a sample S shape to be generated by the spray gun. From the results presented in Table 4, it is obvious that idea of considering compensation in the trajectory is beneficial. With compensation mean deviation is 0.29 mm whereas without compensation it is 1.19 mm. on the same lines, the maximum compensation is only 0.7431 in contrast with trajectory without compensation. This shows that for the considered S shape trajectory, the deviation of obtained trajectory from desired trajectory is considerably less with compensation than without compensation.

This improvement in the performance of the controller is promising for the robot assisted spray painting operation as the error compensation helps to ensure uniform deposition of paint on the surface, accurate trajectory tracking and effective utilization of resources. This methodology will certainly improve the trajectory tracking for the robots used for applications like deburring, implant processing and small component welding which demand for the fine motion planning.

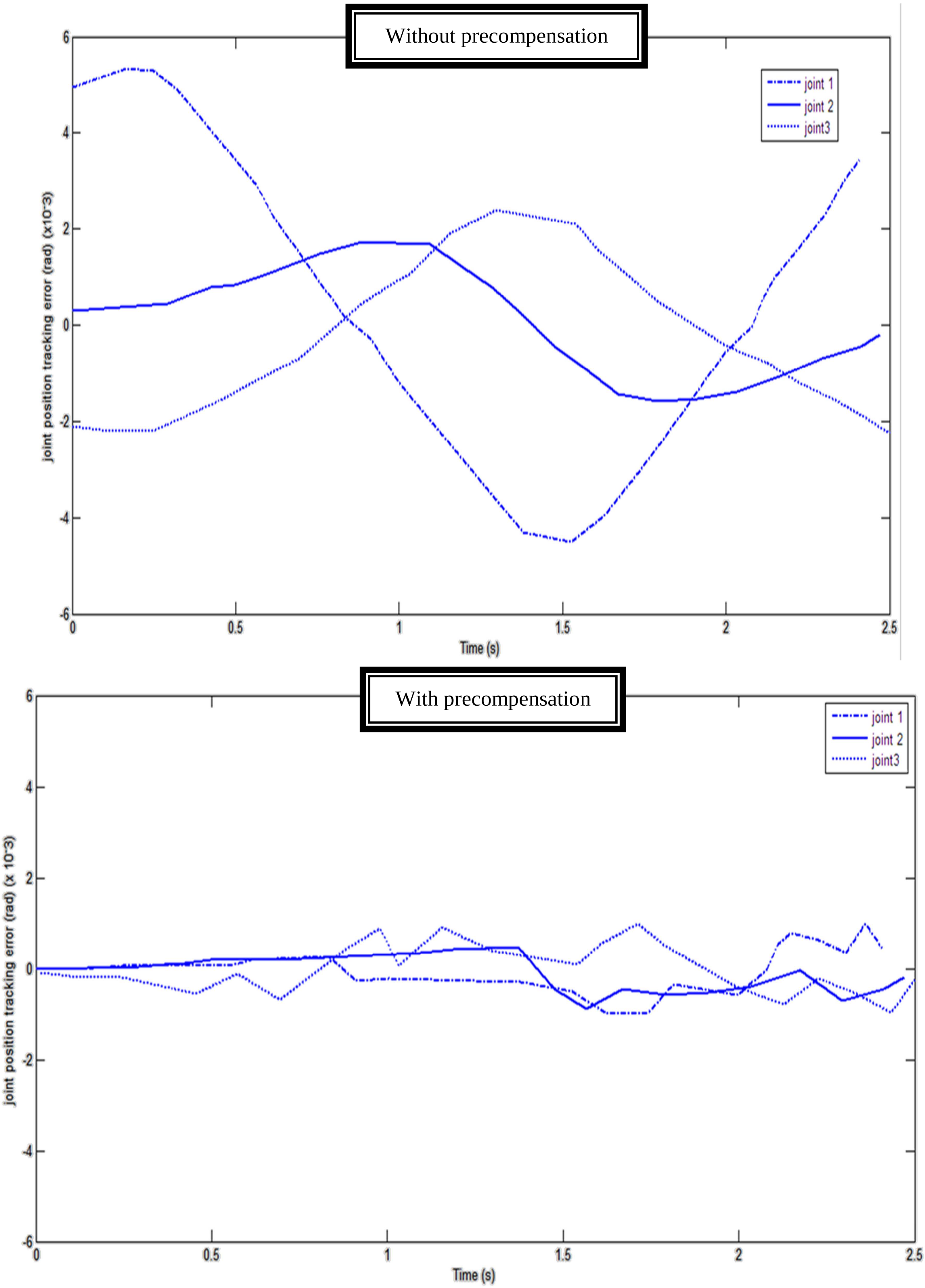

Figure 3 Joint position tracking error without and with precompensation

VI. CONCLUSIONS

This paper explains a novel methodology to compensate the non-linear robot dynamics in an off-line mode. The amount of precompensation is computed for joint angle trajectory by filtering the desired trajectory with an

inverse dynamic robot model including the controller model. The output of the system was the improved velocity feedforward which was integrated to give joint angle configuration. The effectiveness of proposed approach was demonstrated on the simulated robot. The major challenge remains to implement the trajectory compensation on-line on a real industrial manipulator and validate it using arbitrary continuous trajectories generated by for the variety of surfaces to be painted.

REFERENCES

[1] H. Chen , N. Xi, W. Sheng, Y. Chen, General framework of optimal tool trajectory planning for free-form surfaces in surface manufacturing, ASME Journal of Manufacturing Science and Engineering, vol. 127, pp 49-59, 2005.

[2] L. Sciavicco, B. Siciliano, Modeling and control of robot Manipulators, McGraw Hill publications, Newyork, 1996.

[3] Robot Calibration, edited by R Bernhardt. and. S. Albright, Chapman and Hall. 1993. ISBN. 0 412 49140 0. 1993

[4] M. Grotjahn, B. Heimann, Model based feedforward control in industrial robotics, The International Journal of Robotics Research, vol 21, no. 45, pp 45-60, 2008.

[5] Verdonck, W. (2004) 'Experimental robot payload identification with application to dynamic trajectory compensation', PhD Thesis, Katholieke University, Belgium.

[6] S. S. Chiddarwar, Agent based framework for coordinated motion planning of multiple robots, PhD thesis, Indian Institute of Technology Madras, India, 2010

[7] A. Codourey and E. Burdet, A Body-oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots. Proc IEEE International Conference on Robotics and Automation , pp. 1612-1618, 1997

[8] S. S. Chiddarwar and N. Ramesh Babu, Off-line decoupled path planning approach for effective coordination of multiple robots, Robotica, vol. 28, 2010, pp.477-491.

[9] S. S. Chiddarwar, and N. Ramesh Babu, Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach, Engineering Applications of Artificial Intelligence, Volume 23, Issue 7, October 2010, pp.1083-1092.

[10] R. J. Schilling, Fundamentals of Robotics: Analysis and control, Prentice Hall, New Jersy, 1995

[11] Z. H. Jiang, T. Ishada, Trajectory tracking control of industrial robot manipulators using a neural network controller, Systems, Man and Cybernetics, ISIC. IEEE International Conference, pp 2390 – 2395, 2007

[12] C. S. Kim, K. W. Lee, Robust control of robot manipulators using dynamic compensator, Control and Automation, 2009. ICCA 2009. IEEE International Conference, 9-11 Dec., pp 1007-1012, 2009

[13] T.D. Nguyen, O. Egeland, (2005), Output tracking control of a flexible robot arm, Decision and Control, European Control Conference, 44th IEEE CDC-ECC, pp 5269 – 5274. , 2005.